Towards Biomimetic Computing in Machine Vision

Tara Chowdhury, Emmanouela Filippidi, Chris Friel,
Varun Ganapathi, Jenny Liu, Vikash Mansinghka

Abstract

Real-time depth perception and motion extraction remains beyond the reach of machine vision technology except in certain specialized situations, as state-of-the-art general-purpose algorithms are extremely slow and often produce incorrect results. Additionally, we know the human brain can solve stereopsis reliably in real-time using hardware elements operating on the kilohertz timescale in no more than 10-20 serial steps, though we do not understand the computational primitives that enable this. Recent work in computational neuroscience, however, has given us tools which may help us progress towards an improved understanding of low-level perception in both humans and machines. In this paper, we attempt to explain explore the stereovision scheme proposed by Henkel, involving the clustering of local estimates via the synchronization dynamics of leaky integrate-and-fire neurons, with an eye towards both engineering applications and an understanding of neural computing primitives. We demonstrate the plausibility of some of his results, describe our dynamical explorations, and suggest some simplifications and extensions.

Contents

1 Introduction

Real-time depth perception and motion extraction remains beyond the reach of machine vision technology except in certain specialized situations, as state-of-the-art general-purpose algorithms are extremely slow and often produce incorrect results. Additionally, we know the human brain can solve stereopsis reliably in real-time using hardware elements operating on the kilohertz timescale in no more than 10-20 serial steps. This suggests a fundamentally different computational approach to stereovision than is taken by current machine vision algorithms.

Our understanding of how humans produce dense, stable percepts of depth and motion is also incomplete; although we understand how local estimates for these quantities may be produced, we do not understand the computations that yield final percepts. Furthermore, we have only a fledgling understanding of the computational primitives at use in the brain, both at the individual neuron and the neural circuit levels. Such an understanding would dramatically improve our understanding of human perception and may also unlock powerful tools for the engineering of intelligent systems.

Recent work in computational neuroscience, however, has given us tools which may help us progress towards an improved understanding of low-level perception in both humans and machines. Henkel [Hen00] proposed a scheme based on the extraction of local disparity estimates (in a biologically consistent fashion) and the use of a clustering procedure based on the synchronization dynamics of leaky integrate-and-fire model neurons that he claims is capable of producing dense disparity percepts. In this paper, we outline our efforts to explain the poorly understood aspects of his approach and to extend it in several ways.

This paper is outlined as follows. In the remainder of this section, we outline traditional approaches to correspondence problems in low-level vision and motivate the search for novel biomimetic ways of phrasing the computations by comparison to what we know of biological solutions. In Section 2, we discuss some standard biological and artificial models for local correspondence estimators, which form the heart of Henkel-like schemes. In Section 3, we discuss the leaky integrate-and-fire model for spiking neurons, the simulators we built for studying the behavior of networks of these model neurons, and their relevance to clustering problems. In Section 4, we outline the contributions of our explorations so far and list some questions we believe are worthy of further exploration.

1.1 Traditional Approaches to Stereopsis and Optical Flow

The problem of stereopsis is as follows: given two images taken from cameras with some horizontal offset but the same focal point, determine the distance to the nearest object at each point in the scene. The figure below illustrates this problem on two snapshots of a Renault automobile part (where the left camera image is on the left, and the far right image is the depth map produced by an unusual stereopsis approach which produces atypically good results):

The hard subproblem of stereopsis is correspondence - identifying which points in an image come from the same point in the world. Once those points have been identified, determining the 3D coordinate of that point is quite straightforward (given a simple model for image formation, typically involving a thin lens and a pinhole camera, and the application of simple trigonometry). Several excellent references on the depth extraction topic exist; see for example Horn's textbook or the more recent one by Forsythe and Ponce.

Dense correspondence - that is, determining correspondence for all or the vast majority of the objects in the visual field - is particularly difficult because real-world images include large, homogeneous regions (such as the background for the auto part above) with no locally distinctive properties. While sparse methods exist which attempt to identify points in the image corresponding to reliably detectable features and perform correspondence on those features, they typically produce depth information for 1% or less of the visual field. This is clearly too low to be useful in tasks such as obstacle avoidance or object manipulation.

Artificial systems capable of producing dense depth maps from the world would be immensely useful in robotics applications ranging from factory automation to mobile robotics. Methods for obstacle avoidance, general motion planning, object manipulation and recognition could all be improved given depth information. Current machine vision approaches to this problem are quite limited, however.

The only systems which come close to producing dense depth maps from the world in real-time require expensive, special-purpose equipment. A typical example of such a system is a projection-based system, which uses a laser or some other projection device to draw a texture pattern on the world. The texture pattern (if sufficiently varied over the image) can be used to provide distinctive matching information over the entire image; however, projection of such a pattern is an expensive process.

The heart of Henkel's approach to stereopsis is a conversion of the problem to optical flow, the problem of estimating an approximate motion field from a video sequence. In his case, he frames the problem by imagining a virtual camera moving from one eye position to the other; the total flow field induced by the image sequence seen by this camera is the disparity field from stereo. Estimating this disparity field is harder than estimating the flow field, however, since only two samples (corresponding to the two eye positions) are available.

Standard methods for computing dense disparity are very similar in spirit to standard methods for optical flow, and are based on global optimization. An approach to correspondence for optical flow using the calculus of variations for optimization is presented below, based on Horn's seminal 1981 paper.

Let $E(x, y, t)$ be the image brightness at pixel (x, y) at time t. Let $u = \frac{dx}{dt}$ be the x component of the flow and v the y component. Assume the goal is to match contours of equal brightness between frames; this corresponds to the constraint that the "material derivative" of the brightness pattern is zero, or

$$\frac{dE}{dt} = \frac{\partial E}{\partial x}\frac{dx}{dt} + \frac{\partial E}{\partial y}\frac{dy}{dt} + \frac{\partial E}{\partial t} = 0$$

This does not fully constrain the problem. One additional constraint typically adopted is that the flow should be smooth almost everywhere.

These constraints can be incorporated via the calculus of variations as an attempt to minimize the quantities

$$e_s = \int\int ((u_x^2 + u_y^2) + (v_x^2 + v_y^2))dxdy$$

and

$$e_c = \int int(E_x y + E_y v + E_t)^2 dxdy$$

The eventual goal is to minimize a weighted sum $e_s + \lambda e_c$ of these two quantities, where weighting e_c heavily means that the brightness measurements in the images are accurate (and, in a finite realization of the algorithm, the brightness assumption valid). Applying the calculus of variations yields the pair of coupled PDEs

$$\nabla^2 u = \lambda(E_x u + E_y v + E_t)E_x$$

$$\nabla^2 v = \lambda(E_x u + E_y v + E_t)E_y$$

These can be solved by a finite-difference scheme. Simply doing so will not produce good optical flow results, however, as it will attempt to smooth/diffuse the result across object boundaries. A hack to help reduce this problem can be introduced via the finite difference discretization. A standard finite difference technique for a PDE of this form would involve setting each node's value to the average of its neighbors. To help preserve discontinuities, at each iteration the difference between a node and its neighbors could be computed, and if it is above a certain value, the neighbor would not be used in computing the average. Setting the threshold adaptively over time produces reasonable discontinuity enforcement while producing smooth flow everywhere else. A more principled approach to minimizing discontinuities (e.g. actually minimizing the length of line discontinuities) requires optimization techniques like simulated annealing and is therefore computationally impractical.

Other methods which formulate smoothness assumptions (with carefully chosen departures) and attempt to derive matches from this have been developed. Many have been framed in a probabilistic manner, by defining a series of Markov Random Fields, some for disparity and some for the presence of discontinuities, and having their elements interact (where the steady state of the field corresponds to the final disparity field, and is found via belief propagation). They suffer from similar computational problems as the optical flow method mentioned above - finding an equilibrium state is expensive - and often still do not produce good depth maps in certain situations. For an interesting comparison of the performance and running time of various methods, see the Middlebury on-line stereovision database, and note that one leading MRF approach took 30 seconds to execute on a 1.7 gigahertz Pentium machine. This enormous computational cost coupled with our knowledge of neuroanatomy motivates the search for completely novel ways of phrasing the computations involved in stereopsis and optical flow (and perception in general). In particular, the slowness of neural hardware (and the parallelism requirement this imposes) motivates Henkel's approach to both biological and artificial vision.

Before going into the details of Henkel's approach, we outline some of the key differences between traditional computational primitives and those the brain must possess, with an eye towards the larger theme of neural computation.

1.2 Silicon vs. Meatware

Although our models of neural hardware are fairly coarse, they are sufficient to tell us that it operates very differently from silicon hardware from a computational standpoint. Its unique properties seem to be heavily involved in the solutions the brain provides to information processing problems, and may be the key to

finding the right representations and algorithms needed to build artificial solutions. Some of the most salient properties of neural hardware which are different from traditional computing are:

- **Neural hardware is slow; its elements appear to spike on the kilohertz timescale at best.** This is far slower than all modern silicon computers. However, it is able to solve information processing problems such as object recognition in hundreds of milliseconds - far more rapidly than our best artificial solutions. This also suggests that the processing pathways for major computational operations like recognition cannot involve more than tens of steps, as otherwise the delay times would be far too great to account for observed performance.

- **The interconnect in the brain is very loosely specified at "compile time"**. DNA does not contain sufficient informational capacity to precisely determine the wiring of the brain; it is too small by several orders of magnitude. In contrast, the interconnect in silicon computers is specified at great detail by chip designers. Additionally, the software interconnect used in most styles of programming is similarly fairly tightly constrained by programmers; although some programs include flexible dataflow (and have generators for putting the right flows together), this is not characteristic of most programs.

- **The interconnect in the brain changes dramatically during run time.** Instead of specifying precise wiring, the brain appears to contain rules surrounding cell, axon and dendrite growth, which operate over the entire course of an organism's life. These rules define and fine-tune the information processing functions of the brain, giving it tremendous flexibility in the face of major hardware failures. For example, in a particularly compelling series of experiments [SL01], ferrets whose audio and visual inputs were cross-wired experienced cortical changes that reshaped their audio and visual centers to process the new inputs correctly. Most chips and programs, however, exhibit far less reconfigurability, and the majority of their vital functions tend to be quite precisely specified when they are built.

- **The interconnect in the brain includes a far higher local fan-in/fan-out ratio as well as greater diversity in terms of connection length.** Even in VLSI, where the maxim "It's the wires, stupid" commonly holds, neural elements are often far more densely interconnected with their local surroundings, and also exhibit a wider range of connection lengths, with individual cells receiving some local and some extremely long range connections.

- **Individual neural elements are noisy and fail frequently.** Neurons die or malfunction fairly frequently, yet these failures do not significantly impact overall function. They also appear to admit far higher amounts of local noise (in part due to the fundamentally analog nature of some of their computationally significant properties) than typical digital elements do.

- **Neural signal representations and algorithms are massively parallel.** This point is closely related to the previous one; neural elements are noisy and fallible but numerous. Damage to small numbers of neurons in both animals and humans rarely impacts measured performance on particular tasks, implying that processing is somehow spread out among the individual elements.

We believe these hardware-level differences have profound implications on the computational phrasing of perception and action in animals and humans, and could have equally profound and revolutionary impact on robotics if properly understood.

Current work in identifying neural computational primitives has focused on highly abstract models for neurons and neural circuits (typically called "neural networks" in the engineering literature, based on a very rough correspondence between dendrites and multipliers and cell bodies and threshold-and-summing units. This approximation can be justified by assuming a "rate code" model for neural signal processing, where the information output by a neuron is in the form of a spike rate (computed by counting spikes in some fixed time window). While many interesting circuits have been constructed in this way, for example capable of efficiently performing linear algebraic computations in high-dimensional spaces (related to inputs by general nonlinear transforms), we feel these methods represent the tip of the iceberg. Our understanding of neural plasticity from a computational perspective is similarly coarse.

We believe the search for novel computational primitives from the brain (or computational styles or approaches with engineering applications) should involve more detailed representations of the electrical dynamics of neural elements. Henkel's work takes a step in this direction by using the synchronization dynamics of networks of leaky integrate-and-fire neurons to extract "clusters" from pools of noisy estimates for disparity; we will discuss this in further detail below. First, we turn to the front-end of Henkel's system, the disparity estimation mechanism.

2 Local Correspondence Estimation

2.1 Biological Estimators

There are a few definitions that will help us understand the connection between the computational approaches taken to achieve stereovision and the physiology of the visual system. The optical stimulus coming from the outer environment is sensed by a few photoreceptor cells that reside at the retina of the eye. For a small stimulus, the receptive retinal area is on the order of magnitude of a few cells and it has been experimentally observed that stimulation of that retinal area causes increased neuronal response at a particular location at the primary visual cortex. Therefore, the retinal area relevant to the location of neuronal recordings at the visual cortex is termed the "receptive field" of that visual cortex location.

The cells in the primary visual cortex responding to the visual stimuli are categorized as simple and complex. Hubel and Wiesel in 1962 set the basis for the distinction between the two groups and but nowadays there are very precise biological and mathematical descriptions of the two groups. Simple cells have antagonistic rectangular zones that act as ON/OFF receptors, responding to a bar stimulus (one may think of it as a bar of light) of the correct position and orientation only. Maximal response will occur for a bar of the correct length that covers only the excitatory region. Mathematically, the operation simple cells perform is equivalent to linear filtering followed by a threshold nonlinearity. As opposed to simple cells, complex cell respond to moving bar stimuli independent of the bar orientation, but can be specific to the direction of motion of the moving stimulus. Because of their response to a moving bar stimulus, complex cells are equivalent to an edge detector, as different groups of complex cells will respond sequentially to the moving stimulus.

Qian et al [QZ97] have proposed that depth disparity estimators should be extracted given two images from a "left" and a "right" camera situated parallel to each other and thus focused at different points in the horizon at any given time. In any binocular organism, each eye exhibits a different perspective leading to

relative displacements of the objects in the field of view, known as disparities. It is exactly these disparities that allow depth estimation. Therefore, in order to achieve stereovision in a robotic binocular system one would ideally like to compute fast and accurate disparities for each pixel in the two corresponding images. Local disparity estimators that refer to specific pixels are necessary since in one field of view there can be objects located at different depths.

Qian et al described an algorithm for estimating disparities based on the notion of a stationary bar using simple cell responses. The best way to detect the edges of a light bar stimulus is to convolve the signal with a Gabor function (multiplication of a Gaussian envelope with a cosine), which in the Qian case is a 2D Gabor, as opposed to the Henkel algorithms, where everything he proposes is in 1D. (This is consistent with Ohzawa's recordings, in which a Gabor response closely modeled the receptive field profile of simple and complex cells).

A 2-D Gabor function with mean 0 and horizontal cosine orientation has the following form:

$$G(x, y, \omega, \sigma_x, \sigma_y, \phi) = \frac{1}{2\pi\sigma_x\sigma_y} e^{\frac{-x^2}{2\sigma_x^2}\frac{-y^2}{2\sigma_y^2}} cos(\omega x + \phi)$$

We note that the Fourier transform of a 1D Gabor function is a pair of Gaussian bumps whose center frequency is that of the Gabor's cosine and whose bandwidth is inversely related to the spatial extent (σ) of the underlying Gaussiam; this helps to understand their action as linear filters, and the importance of a proper selection of scales to include all information in the input images.

In Qian's setup, a simple cell consists of four Gabor filters, two in the left image and two in the right. The two Gabors for each image are 90 degrees out of phase, while the difference between Gabors between the left and right images is $\Delta\phi$. The complex cell responses are obtained by convolving these filters, squaring the responses, then summing them up. Qian has shown empirically (and confirmed by Taylor-series analysis) that this yields an approximate preferred disparity of $\frac{\Delta\phi}{\omega}$ - that is, in an ideal situation, a complex cell's response is maximized if the real stimulus disparity at the center of its simple cells' receptive fields is equal to the ratio mentioned above.

A complete disparity estimator is constructed out of a group of Qian estimators with different $\frac{\Delta\phi}{\omega}$s by reporting the estimate corresponding to the $\frac{\Delta\phi}{\omega}$ for the maximally active complex cell. In Qian's early experiments, he used a single estimator of this form per pixel, and yielded extremely noisy disparity maps. We note that a natural limitation on the range of these estimators stems from the periodicity of the cosine function; other limitations have been derived, but we do not explore them here.

The estimators Henkel used in his paper are actually based on the opponent-energy motion estimators of Adelson and Bergen [AB85]; although they use quadrature-paired Gabor filters, they produce disparity estimates by performing arithmetic on their response rather than by taking a maximum for several phase values. We do not detail them here, but note that this makes them far computationally cheaper at the cost of a vastly reduced working range. Qian's work seems to suggest his estimators are better models for the detectors actually present in the brain.

We note that the Gabor functions discussed here can also be used for phase-based estimation of correspondence, according to the Fourier Shift Theorem: if one assumes that two images are locally related by a fixed shift, this should be detectable by a difference in phase of their local Fourier transforms. The phase of a Gabor filter's response can be used in this manner to compute an approximation for disparity; this approach

was first proposed by Sanger [San88] and has been improved upon in many ways since. While this is not precisely the way in which Gabor filters are being used by Qian, we feel this provides some insight into the relevance of a Gabor-type filter for correspondence estimation.

2.2 Artificial Estimators

Several local estimators of correspondence have been proposed in the machine vision literature.

2.2.1 Correlation-based Block Matching

The most commonly used artificial disparity estimators involve local block-matching based on a normalized correlation measure. The idea is to take a window around each pixel in one image and search for the offset in the next image with maximum normalized correlation. In software, this is frequently implemented in the frequency domain; in hardware this can be efficiently implemented in parallel. The few available commercial real-time stereo systems operate in this manner (e.g. those provided by Videre Design), although they frequently produce incredibly inaccurate results, with a strong dependence on window size choice, and are limited to integral disparities.

2.2.2 Gradient-Based Estimators

The optical flow description provided earlier suggests another scheme for local disparity estimation. Apart from the incomplete "brightness conservation" constraint, one can stipulate that locally (in some small window, perhaps weighted) the disparity must be constant (in a manner similar to some popular variations on an approach proposed by Kanade). While these estimators are also limited to integral pixel disparities, they can be efficiently computed via a few convolutions and multiplications, and can easily incorporate pre-shift biases of the sort used by Henkel's estimators. A detailed exploration of these estimators (and the knowledge that they reliably yield clusters) would be extremely valuable, as they are likely candidates for machine vision systems based on Henkel's ideas.

We also note that local estimators based on analyzing the principal texture direction in image patches can also be constructed, following [BGW91]; the computation boils down to identification of a spike in the *local* Fourier transformation of the image. The details of these estimators are beyond the scope of this paper, but they remain another candidate for an artificial vision system.

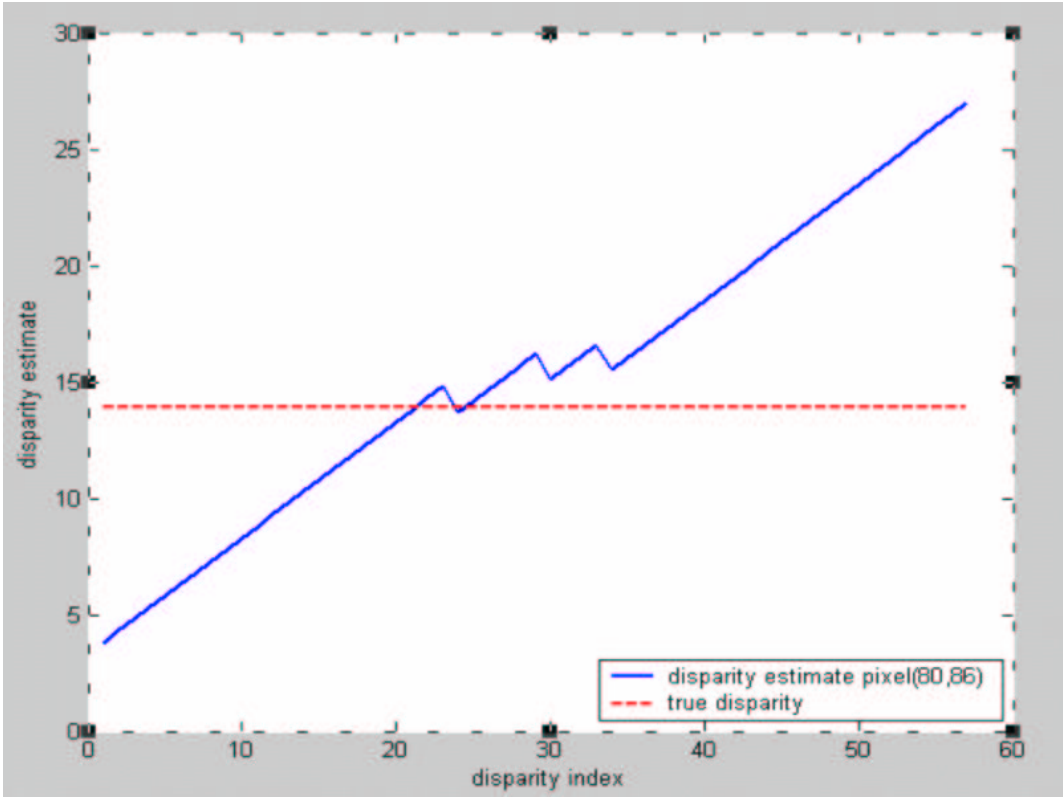
2.3 Forming a Coherent Percept

Henkel's approach hinges upon the following notion: rather than using a single Qian-style estimator (with resolution determined by the sampling of $\Delta\phi$) or Adelson-style estimator (with small working range), a bank of estimators should be used with different biases. His hope is that those estimators whose working ranges include the true disparity will all roughly agree, while the others won't be correlated.

Biasing Adelson and Qian-style estimators is quite straightforward: one simply applies a preshift to the underlying Gabor function, or (essentially equivalently) shifts the mean of the Gaussian and the phase of the cosine appropriately. Qian has shown that position-shifted Gabor-based complex cell estimators have different error properties than purely phase-based ones, and that a combination of position and phase shifts

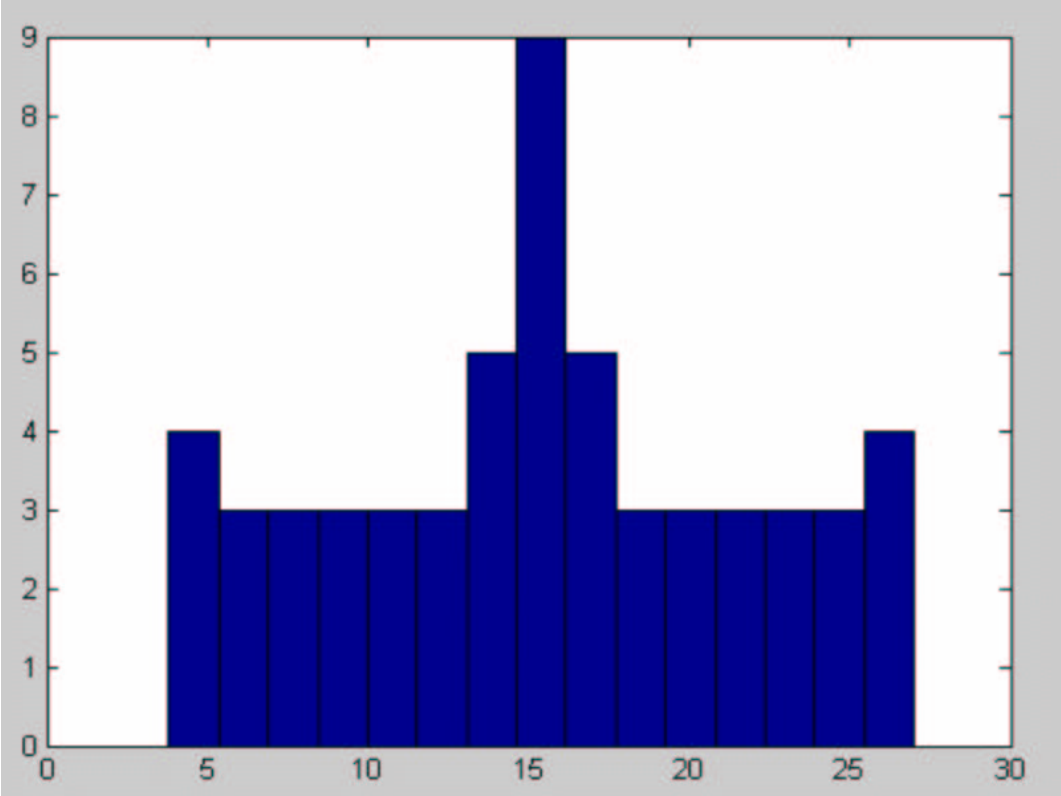
yields the highest accuracy. Henkel's scheme assumes this property for all estimators, again hoping that erroneous estimates will appear random as compared to correct ones.

We implemented banks of Qian and Adelson-style estimators and tested them on standard stereo images with known ground truth, such as the "map" image from the Middlebury College Stereo Vision Resesarch Database[1]. We show the results of some runs of Qian- and Adelson-style estimators on these images. In each case, we tried a wide range of preshifts (determining the effective range of our overal algorithm).

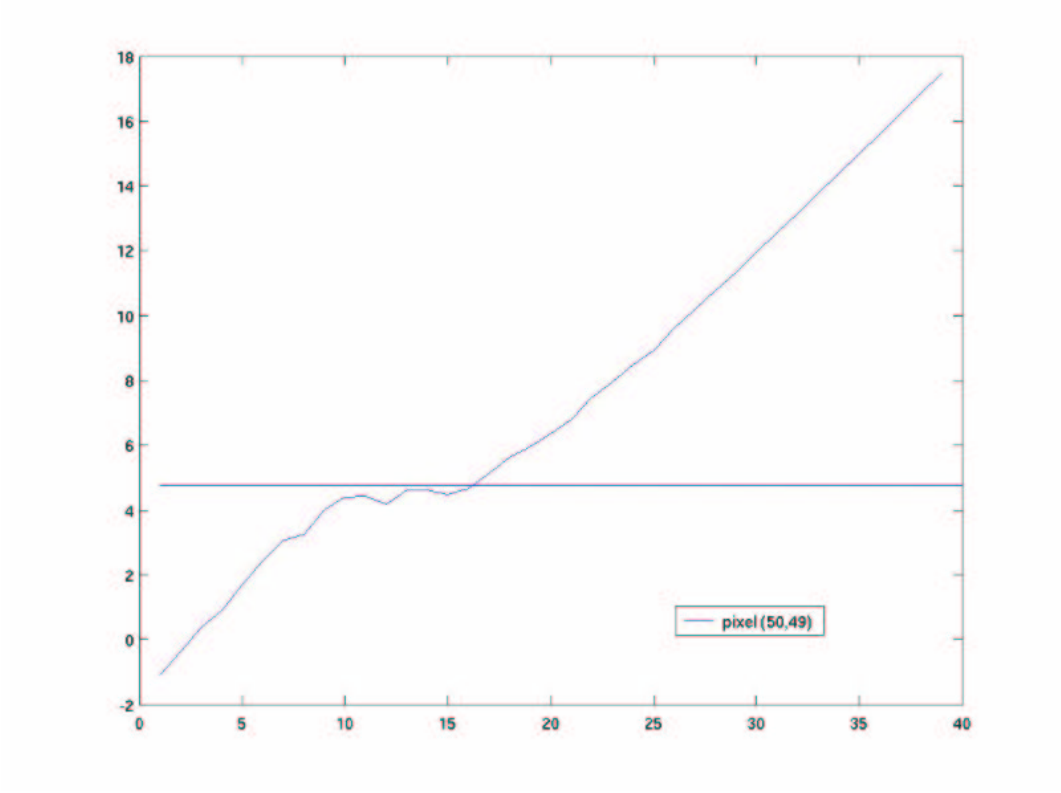


In the figure above, the y axis shows disparity estimate, and the x axis is a linear function of the bias of the underlying disparity estimator (set by the preshift of its component Gabor functions). Note the agreement around the correct disparity. The jaggedy response is likely due to the coarseness with which we sampled $\Delta\phi$. A histogram of the estimates is shown below:

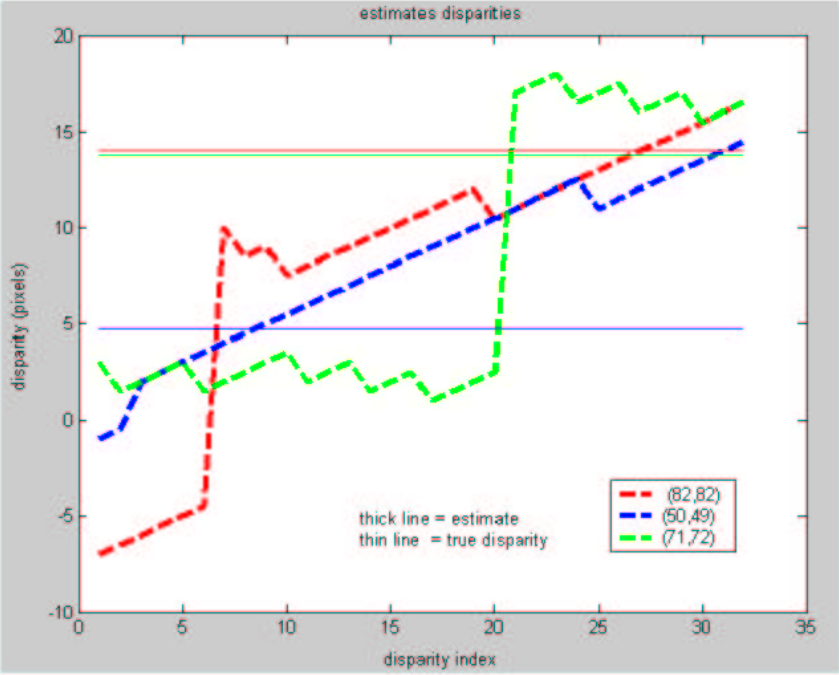
[1] Available online at `http://cat.middlebury.edu/stereo`



Below, we show a similar plot for an Adelson-based estimator, and again note the agreement near the correct disparity value:



Unfortunately, these results are *not completely representative*. In the figure below, for example, we note that none of the 'consitent' regions correspond to the correct disparity, as evidenced by the ground truth:



Similar plots were obtained for some pixels using the Adelson-based estimators. We believe the disconnect between Henkel's claim that clusters form for Adelson-based estimators at minimum (and his conjecture that they form reliably for other estimators) is due to differences in estimator tuning, yielding a bias towards images with specific frequency-domain profiles. The positive examples, however, serve as partial validation of Henkel's core stereopsis idea.

We now return to the underlying comptuational question. Given a pool of noisy estimates, there are several options for producing stable disparity percepts. For estimators that produce actual disparities (as opposed to responses, ideally peaked at particular disparities), some sort of voting scheme (in the spirit of Henkel's approach) is plausible.

Many artificial algorithms consistent with the high-level term "voting" exist. The simplest one involves histogramming: simply form a histogram with a fixed bin-width for each pixel containing all its disparity estimates, and report the disparity corresponding to the fullest bin. As seen in the examples above, this approach is likely to fail when estimates might naturally be split between two adjacent bins. This might be alleviated by a "sliding histogram": choose a fixed bin width based on the expected agreement range of the disparity estimators being used, then consider a series of bin boundaries consistent with that width, effectively sliding the histogram along until it aligns with an offset of an entire bin width. The reported match in this case would correspond to the average of the members of the fullest bin across all bin shifts.

Henkel advocated an approach which is more sophisticated, at least on the surface: attempt to detect "clusters" in the disparity estimates. Many algorithms (k-means, spectral clustering, etc) exist for identifying

"clusters" in data; each adopts a particular mathematical criterion for forming cluster boundaries, and provides an algorithm (typically somewhat computation intensive) for actually choosing the boundaries.

Henkel's particular approach to clustering relied on the synchronization dynamics of certain models of neurons; as the synchronization dynamics of these models is not understood and the structure of partially synchronized states cannot be written down in a closed form, this does not constitute a computational definition of the clusters he intended to extract. Without a better characterization of the expected cluster size and shape (or at least diameter) of the clusters produced by the local estimators of choice, it is impossible to determine what algorithm is most appropriate in this case, but we believe that if disparity estimates cluster as advertised, a wide variety of approaches should suffice, if properly tuned. The choice of a particular algorithm should then depend on the constraints of the hardware in question; if meatware, synchronization dynamics of some kind of neural model seem a viable candidate, but if real-time stereo for robots is the issue, an approximation based on histogramming should theoretically suffice.

We experimented briefly with histogram-based approaches at identifying clusters, but did not achieve good results over many images, likely due to the tuning errors in the disparity estimators we were using.

3 Synchronization of Spiking Neurons

In Henkel's system, the per-pixel estimates produced by the estimators outlined in the previous section were fed into a network of simple nonlinear elements based on a standard, simplified model of a neuron, known as a *leaky integrate-and-fire* neuron.

3.1 Leaky Integrate-and-fire Neurons

A leaky integrate-and-fire element s is described by the following rules:

- $\frac{dV_{c,s}}{dt} = \frac{-V_{c,s}}{RC} + \frac{I_s(t)}{C}$
- $\frac{dV_{o,s}}{dt} = \frac{-V_{o,s}}{K_{out}}$
- If $V_{c,s} = V_{th}$, then $V_{c,s} \leftarrow 0$ and $V_{o,s} \leftarrow V_{init,s}$
- $I_s(t) = K_s + \Sigma_{s' \neq s} \epsilon V_{o,s'}$

Here the index s identifies units in a group connected by an all-to-all coupling (without self-interaction) of strength ϵ.

A simple equivalent circuit model for such an element consists of a resistor R and capacitor C in parallel charged by a current source K_s, where the capacitor's voltage is read by a comparator whose other input is a voltage source V_th and whose output activates a discharge transistor across the capacitor. The leakiness of the unit is due to the resistor R; in the absence of any input current, the capacitor's charge will leak to ground. Its integrative action is easy to see by considering the transfer function (Laplace transform, if you like) of the circuit.

This model captures some of the phenomenological properties of the electrical behavior of real neurons. A detailed discussion of the connection between this model and more physically accurate neural models (such

as the Hodgkin-Huxley model) is beyond the scope of this paper; see e.g. [Izh04] for an interesting dynamical taxonomy of the behavior of the Hodgkin-Huxley model and many of its simplifications.

Note that in the absence of coupling and assuming zero initial V_c, V_c evolves as:

$$V_c = RK_s(1 - e^{-\frac{t}{RC}})$$

so

$$t_{spike} = ln(\frac{V_{th}}{RK} - 1)RC$$

Also note that if V_{th} is less than RK, the element will never fire. Henkel defines ϵ to be w_{CC}/N_s where N_s is the number of synchronizing elements per pixel. The idea here is presumably to keep the total input current for each element roughly constant regardless of the number of disparity estimators per pixel. Henkel kept $w_{CC} = 0.7$ in his simulations.

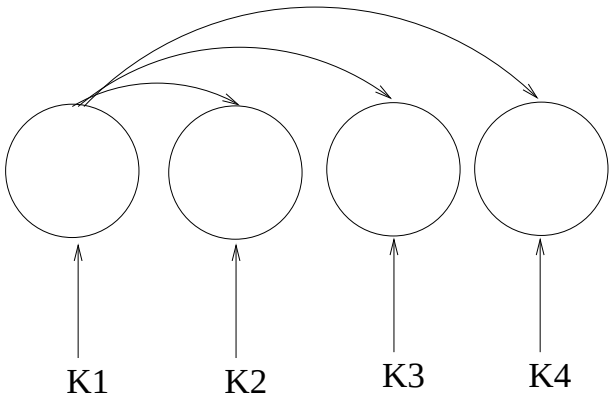
Henkel actually used leaky integrate-and-fire units with a fixed refractory period; this means that after a spike is fired, the unit's internal potential V_c is fixed to 0 for a fixed time interval t_r. The significance of this will be discussed below.

3.2 Clustering via Partial Synchronization

Henkel claimed that the computational purpose of his networks was "clustering". The idea was that disparity estimates would be converted into current drives for a bank of leaky integrate-and-fire neurons, each all-to-all coupled. Those elements with sufficiently compatible natural frequencies (determined by their current drive) would synchronize; if this synchronization reliably occurred, it would group the disparity estimates into clusters.

This effect is grounded in the dynamical study of these units. It has long been known that pulse-coupled oscillators of the same frequency but different initial phases can synchronize (even stably), and similarly that pulse-coupled oscillators of different frequencies can synchronize in groups. The analytical work in this area originated with A. Winfree; a recent extension to his framework can be found in [AS01]. Unfortunately the conditions determining the presence of partially locked states are still poorly understood and virtually nothing is known about their structure; the only useful pointer we were able to discover was in [Bot96], in which numerical experiments along with a qualitative explanation of synchronization of units with different frequencies suggested that a refractory period was a crucial enabler for synchronization of units with different intrinsic frequencies.

A precise description of Henkel's synchronizing layer is as follows. Assume we have a series of disparity estimates $d(i, j, s)$ for each pixel (i, j). For each pixel, construct a series of leaky integrate-and-fire units with refractory period (set $V_th = 16, R = 40, C = 0.00625, t_r = 0.2, K_{out} = 100$). Call the group of these s's per pixel the "coherence layer" or "disparity stack" for pixel (i, j). Schematically, this network looks like:



Each circle represents an element fed by constant input $K_s = 9.5 + 0.65d(i, j, s)$. The coupling is actually all to all without self-interaction, but the other arrows aren't shown due to messiness. The linear function relating disparity estimates to currents presumably tunes the effective clustering range of the network; we believe Henkel must have set these parameters by an optimization procedure.

We note that Henkel's choice to encode disparity estimates in frequency is not the only one imaginable; phase seems the most natural alternative, and might permit easier analysis. We could not find useful leads on the stability of partial phase-locking, however, so we did not pursue this avenue further.

3.3 Simulation of LIF Neural Networks

Leaky integrate-and-fire neurons can be simulated economically in at least two ways: via a simple first-order Euler approximation to the underlying differential equations, and via an analytical approach which assumes the output spikes are transmitted instantaneously. We implemented an Euler simulator and drafted a time-to-spike simulator.

The idea behind a time-to-spike simulator is as follows: each element's dynamics *between spikes* is described by a simple first-order differential equation, given a constant input current. Using the solution for t_{spike} mentioned above and given the integral of the impulse produced by a unit (and fed to all others) during a spike, then, an essentially exact simulation can be performed: simply maintain a list of the times to spike of each unit, adjusting it after each incoming spike based on the impulse received. (The refractory period can simply be added to the time-to-spike; this adds no complexity.) In Henkel's case, the spikes were decaying exponentials with initial value 1; given the membrane capacitance, this yielded an impulse of area $1.6V$ per spike.

Alternately, an Euler approximation for these elements is appropriate because the discontinuities produced by the threshold-reset spiking behavior of these elements cannot be tracked with high precision; higher-order approximations and adaptive timestep schemes will therefore waste a tremendous amount of effort and many introduce substantial inaccuracies near the many (and vitally important) discontinuities in the system. The approximation itself is simple enough that we will not detail it here; we validated it by testing various timesteps and selecting one which yielded a time-to-spike consistent to 99.9% with analytical predictions.

The Euler simulator incorporates the time course of a spike into the model at the cost of numerical inaccuracy. It was not clear to us from Henkel's work whether or not he took this into account; in principle,

however, it seems like this should not be significant, as the spikes are short enough relative to the other timescales in the system to be decent approximations to delta functions. Regardless, most of our experiments were conducted with the Euler simulator, as we implemented it first. Simulations were carried out via a special purpose engine written in C++ with a MATLAB interface for convenience.

3.4 Synchronization Experiments

We conducted a series of numerical experiments to explore the structure of partially synchronized states. Our goals were as follows:

1. Identify an approximation for the maximum frequency separation between two LIF elements consistent with their synchronization, where synchronization is defined as the co-occurrence of spikes (spike times within some small epsilon of each other) for all times greater than some synchronization-times.
2. Study how the size of that maximum separation changes, if at all, as more elements are added.
3. Attempt to predict synchronization by e.g. spectral clustering on the graph of frequencies with edges consistent with the approximate synchronization distance found above.

We encountered a variety of intriguing behaviors in the system that we are unable to explain. For example, when using units with Henkel's parameters, we found that the frequency separation varied with the frequency range. For example, at frequencies corresponding to current drives of 10, a separation of 0.22 in drive was sufficient to prevent synchronization. Near drives of 20, however, a separation of 0.24 was insufficient. We also found that shifting from e.g. 2 to 3 elements altered the maximum permissible drive offset; near a drive of 10, for example, three units with drives of 9.78, 10.0 and 10.23, would reliably converge.

As we were unable to characterize this behavior in any sort of parameter-independent way, we cannot provide bifurcation diagrams for this behavior. Future experiments where oscillation frequency ratio (instead of absolute drive difference) is explicitly controlled may reveal a more uniform structure. We did find that increasing the refractory period increased synchronization range in some cases, supporting the work of Bottani, but were not able to conduct a systematic study of this phenomenon.

We attempted to assess the suitability of synchronization dynamics at executing clustering computations; the only verification of this we were able to provide was based on constructed input distributions. For example, on the few tests of this we performed, if we fed the units with samples from a gaussian of small variance, they would frequently synchronize to a value near the mean of the gaussian. We were even able to reproduce this behavior in the presence of many (50%) uniformly distributed distractors. We do not report plots for exact ranges because we were unable to get a handle on the underlying scale parameters of the system (that is, a reasonable relationship between the maximum supported gaussian variance and the element parameters). A more robust analysis of this from an estimation theory standpoint however, may be a useful contribution to our understanding of neural signal processing.

The only other observation we made worth nothing is that our numerical simulations revealed an odd time-course of synchronization for small networks. We outline it for the $n = 2$ case. Consider two units whose intrinsic frequencies had been empirically determined to lie too far apart for synchronization. The effective frequency implied by the time between spikes for the unit with lower frequency would typically rise to a peak (with vaguely exponential shape) then decay back down to its natural frequency. We are

not sure of the dynamical significance of this, though we feel it would be worth investigating further, by e.g. determining if the curves are in fact exponential, and if there is an easily understandable relationship between the constant and the parameters of the network in question.

We now return to Henkel's overall system.

3.5 Henkel's Read-out Approach

Henkel detected synchronization using another network of spiking units of identical organization and parameters as those above; the only difference was that this layer received input currents equal to the sum of the output currents coming from the previous layer. The intuition is that he was using the spiking units as coincidence detectors; by tuning the threshold, the read-out units could be made to fire if a specific number of coherence layer units fired within a short time interval (related to the leakiness R).

If a single cluster was present in the coherence layer, then, the read-out layer should spike at an equivalent frequency. It is worth noting that the read-out layer could be replaced with a single coincidence detector for each disparity stack; the extra units actually reduce accuracy, by artificially inflating read-out spiking frequency. Henkel's read-out scheme also represents prior knowledge about cluster size and is vulnerable to error in the presence of multiple clusters.

From an engineering perspective, a single read-out neuron should be used, whose thresholds are tuned to the expected cluster size; if multiple clusters are expected (e.g. in the case of transparency), a more elaborate read-out scheme would be necessary. One option would be a series of read-out units tuned to different cluster sizes, selecting the output from the most demanding unit. While this could still be fooled by multiple clusters of the same size, it may be a step in the right direction.

While a naive application of Henkel's approach yielded a reasonable disparity estimate for a few test pixels with reasonable disparity, we were not able to reproduce overall percept formation reliably; again, we believe the primary problem to be with our implementation of a Henkel-like front-end.

3.6 Other Interesting Computations

It is worth noting that a variety of other computations have been implemented using the synchronization dynamics of leaky integrate-and-fire units, though in virtually all cases the computation remains exceedingly poorly understood. The best example of this is Choe's contour completion and segmentation system [CM03], which uses synchronization dynamics to assemble clusters of contours based on orientation agreement of local line segments. Choe's connection topology is slightly different from Henkel's (consisting of locally excitatory connections and long-range inhibitory ones, equally poorly understood from an engineering standpoint), but the underlying computational semantics seem quite similar. One difference worth pointing out is that Choe uses phase (as opposed to frequency) synchronization, suggesting that computation using phase synchronization is possible; it remains unclear to us whether this is actually easier to understand.

We mention his system and allude to others so that the reader will keep in mind the general applicability of the principles under study, and look beyond the context of stereopsis and optical flow in which we are exploring them.

4 Contributions and Future Work

4.1 Human Disparity and Motion Perception

We have verified that under good conditions, estimators based on the most widely accepted models for complex cell function (Qian's hybrid phase and position estimators) do exhibit clustering behavior. To our knowledge, this does not appear (for Qian-style estimators) anywhere in the literature, nor does the expected cluster size for Adelson/Henkel estimators. This suggests that a Henkel-like model for disparity percept formation is reasonable. However, we have not yet been able to demonstrate reliable cluster formation with Qian-like estimators (verifying cluster presence by visual inspection), and therefore we have not been able to propose an algorithmic characterization of the process needed to extract reliable estimates.

We believe these difficulties are due to the particular parameter tunings we have adopted and the corresponding frequency-domain limitations of our estimators. The underlying concept of the model, the use of local estimators in a voting process to produce reliable and dense disparity estimates for both real images and sparse random-dot stereograms involving transparency, remains viable.

The most important future work needed in this area involves tuning Qian-like estimators according to our best psychophysical estimates and assessing the viability of clustering, and comparing the limits of cluster stability with human limits (e.g. sparseness of random-dot stereograms and image frequency content vs human and Qian-cluster stability).

4.2 Machine Vision

The embarrassingly parallel architecture for correspondence suggested by Henkel may free vision engineers from solving difficult optimization problems on a frame-by-frame basis. Our partial results suggest that Adelson-style estimators are likely to be more suitable for disparity and motion percept formation than Qian-style ones in artificial contexts, as they require far fewer convolutions per pixel. However, we believe the clustering behavior observed should also occur for gradient and texture-orientation-based estimators, and that a machine implementation would likely achieve better performance using these due to their computational simplicity.

We also believe that a simple voting procedure (such as a sliding histogram approach) should be sufficient to extract disparity estimates from the estimators' outputs. The combination of this approach with gradient-based estimators should permit an implementation on standard hardware; one option would be to have two processors, one performing estimate extraction and one performing histogramming, with the cameras connected to the first processor via a bus with capacity comparable to FireWire S100 (100Mb/sec) and with both processors having access to a few megabytes of shared memory with no strict locking requirements (enough to store roughly 50 estimates per pixel). Assuming 1k FLOPS per pixel to estimate (a vast overestimate for gradient-based estimators) and 1k FLOPS per pixel to vote (an overestimate for simple histogramming, where counts can be updated by division by binsize, and the final value found by a linear search), two 1GHz processors should be able to handle several frames per second for 256x256 images[2]. A specialized circuit implementation could almost certainly do far better, exploiting the parallelism, though

[2]These come from simple arithmetic for a rough lower-bound on the number of frames, and depend upon the images fitting into cache

this may not be necessary for e.g. robotics applications.

While an analog implementation of a network of spiking neurons could likely be used to extract estimates, it seems likely that a simpler approximation implementable using COTS hardware (or even standard digital logic building blocks) would be more cost-effective in engineering applications. The constant cost per pixel provides a substantial improvement over optimization-based stereo algorithms and may be sufficient.

The next steps in this area involve implementing gradient and texture-orientation-based estimators and measuring cluster stability, as well as assessing the performance of histogram-like voting algorithms for identifying clusters on benchmark stereo images.

4.3 Neural Computing Primitives

Our numerical explorations suggest clustering can be performed via the synchronization dynamics of leaky integrate-and-fire neurons, though we do not understand the process deeply enough to enable an engineering characterization. Additionally, our qualitative analysis suggests that Henkel's read-out layer is unnecessarily complex, identified its inflexibility with respect to the number of units in a cluster, and yielded a proposal for a simpler read-out system.

Future numerical experiments may shed more light on synchronization as well as validate our read-out approach. However, we believe further dynamical analysis is essential before real progress in this area can be made. We believe one promising approach to understanding the clustering properties of leaky integrate-and-fire neurons is through A. Winfree's work on pulse-coupled oscillators; our hope is that a fresh look at that proof (along with modifications) might yield good approximations for finite numbers of oscillators and insights into the structure of partially synchronized states. Near the onset of synchronization, a group of weakly coupled spiking neurons should behave like a network of coupled limit-cycle oscillators; we hope that dynamical analyses along these lines will yield computational insights in the future.

We note that the search for neural computing primitives has been particularly hampered by our lack of understanding of neural data representations and the emphasis in neuroscience on identifying "functional regions". From a reverse engineering standpoint, this is somewhat analogous to searching for the high-level blocks in a block diagram without any understanding of exactly what information is transmitted along the wires. Individuals experienced with reverse engineering in the domain of electrical (and computational) systems will appreciate the backwardness of this approach: it is often far easier to infer circuits from signals and algorithms from data structures than it is to go the other way around.

We hope that continued investigation into the electrical dynamics of realistic neuron models will shed light on what their wiggling potentials actually mean and enable the useful identification of well-characterized computational primitives.

4.4 Conclusion

Our results so far suggest the promise of Henkel's approach to stereovision and hint at its applicability to optical flow. However, a host of questions remain to be answered before methods like Henkel's can be broadly adopted. In addition, the computational possibilities afforded by leaky integrate-and-fire (and even more biologically accurate) neuron models are only beginning to be explored, both as models for biologists and inspiration for engineers.

Acknowledgements

We would like to acknowledge Dr. Rolf Henkel for useful comments on disparity estimation, Prof. Erik Winfree for encouragement throughout CBSSS, and Prof. Andre' DeHon for dragging us all out to Pasadena to think about interesting things.

References

[AB85] E. H. Adelson and J. R. Bergen, *Spatiotemporal Energy Models for the Perception of Motion*, J. Opt. Soc. Am. (1985).

[AS01] Joel T. Ariaratnam and Stephen H. Strogatz, *Phase Diagram for the Winfree Model of Coupled Nonlinear Oscillators*, Physical Review Letters (2001).

[BGW91] J. Bigun, G. H. Granlund, and J. Wiklund, *Multidimensional Orientation Estimation with Application to Texture Analysis and Optical Flow*, IEEE Transactions on Pattern Analysis and Machine Intelligence (1991).

[Bot96] Samuele Bottani, *Synchronization of Integrate and Fire oscillators with global coupling*, eprint arXiv:cond-mat/9607056 (1996).

[CM03] Yoonsuck Choe and Risto Miikkulainen, *Contour integration and segmentation in a self-organizing map of spiking neurons*, Biological Cybernetics (2003).

[Hen00] Henkel, Rolf, *Synchronization, Coherence-Detection and Three-Dimensional Vision*, Technical Report of the University of Bremen (2000).

[Izh04] Eugene M. Izhikevich, *Which Model to use for Cortical Spiking Neurons?*, IEEE Transactions on Neural Networks (2004).

[QZ97] N. Qian and Y. Zhu, *Physiological Computation of Binocular Disparity*, Vision Research (1997).

[San88] T. D. Sanger, *Stereo disparity computation using Gabor filters*, Biological Cybernetics (1988).

[SL01] M. Sur and Catherine A. Leamey, *Development and Plasticity of Cortical Areas and Networks*, Nature Neuroscience Reviews (2001).